Byte_order_conversion_example.py

```
ip_address = "1.2.3.4”

# inet_pton converts an IP address string in dotted quad
# (presentation) format, to a bytes object in network byte
# (big-endian) order.
ip_address_bytes_network = socket.inet_pton(socket.AF_INET, ip_address)
print(ip_address_bytes_network)
```

```
b'\x01\x02\x03\x04'
```

```
ip_address_bytes_network = b'\x01\x02\x03\x04'

# Convert the IP address bytes object back into an int, in
# big-endian byte order.
ip_address_int = int.from_bytes(ip_address_bytes_network, byteorder='big')
print(ip_address_int)
```

```
16909060
```

```
ip_address_int = 16909060

# Print it out the way the little-endian machine stores it.
print(int.to_bytes(ip_address_int, length=4, byteorder='little’))
```

```
b'\x04\x03\x02\x01'
```

```
ip_address_int = 16909060

h = socket.htonl(ip_address_int)
print(h)
```

```
67305985
```

(host is little endian)

```
h = 67305985
i = int.to_bytes(h, length=4, byteorder='big')
print(i)
```

```
b'\x04\x03\x02\x01'
```

```
h = 67305985
print(int.to_bytes(h, length=4, byteorder='little’))
```

```
b'\x01\x02\x03\x04'
```